fwIntro
00.01.29 (d6

To Ted Nelson Home Page

FLOATING WORLD™
A NEW USER ENVIRONMENT

NO ICONS • NO METAPHORS • NO FILES • NO APPLICATIONS

For the last twenty years, as the personal computing world spilled forth, I have been increasingly indignant about all of it.

The sheer impossibility of everything I wanted to do with computers filled me with rage. The clumsy unconnected windows! The tyranny of "applications" owned by different software manufacturers! The clumsy one-step "undo"-- you can undo only one operation, and only in an application, not in the operating system! The prisons of "files" and their naming! (Conventional computer file structures have created a horrible world, and nobody realizes it, nor scarcely perceives the mess of contradictory workarounds they have engendered.)

But there is no way to get a foothold without BUYING IN-- without accepting somebody else's definition of the world. The Macintosh, Director, C++, HTML, you name it (even Unix)-- all are clumsy, and in my view, twisted tools that won't do what I think is important. The fundamental interface, designed at Xerox PARC in 1974, has not improved since that time, and has become a prison. And one of the tragedies of our time has been "word processing", especially as tangled with clumsy filing, clumsy naming, and the "clipboard"-- the computer guys thought text was simple, and therefore gave the world woefully inadequate tools.

The real problem was that we need a change at the operating system level.

Where to begin?

As Archimedes is quoted as saying, "Give me a lever long enough and I will move the world." And, of course, a fulcrum.

THE FULCRUM

I always thought that a system I designed in the 1980s, ZigZag™, had promise. It is a construct universe that lets you build any structure fairly easily, and also see it. It consists of multidimensional list structures with built-in row-and-column visualizations.

However, it took a long time to get ZigZag implemented. But now it has been prototyped and proven. In 1997, with the help of Andrew Pam of Xanadu Australia, we got a prototype running, and it has proven to be what I had imagined and visualized.

And now, with the help of Tuomas J. Lukka at Jyväskylä University in Finland, perhaps we can begin again.

STARTING OVER

I have designed a new computer universe from scratch. Prof. Lukka says he will build it.

Floating World is the name of the environment.

ZigZag is the kernel. (We will assume that you are familiar with ZigZag. If not, there is a tutorial.)

All structures are implemented in ZigZag. The user doesn't necessarily see that structure unless interested, but it's right there if you look under the hood.

Nearly everything is composed of zzcells, zzlists and permascrolls (explained below). Occasionally we also have to use conventional files, for conventional data structures which already are locked in.

ELEMENTS OF FLOATING WORLD

I. STRUCTURES

The fundamental unit is the ***thinglet***. It is not an object, an entity, a file, a table, a record, or those other nouns whose computer meanings are already committed. It is something different. Therefore a different name.

A thinglet is composed of ZZcells. It has a ***maincell***. A maincell is that cell from which the path to any of its other functional cells may be defined. The maincell has a ***handle***, free to be listed and arranged in all dimensions except d.handle, along which it attaches to the maincell.

All the internal operations of Floating World are based on thinglets, their handles and maincells, and paths between various significant cells of the thinglets.

The fundamental user-significant unit is the ***item***. An item is anything which interests a user, including: • free-floating notes • sticky notes • projects • documents • versions • graphics. Items may have many different user-definable structures, but they are fully reachable through ZigZag visualizations and construction methods.

(TEXT)

Text must be simple for the user, but text operations will not be simple internally, and items consisting of text may be quite complex. This is because we afford full Xanadu® connectivity to all text, even titles (see below). We relegate text to being a special case of fluid media (below).

The fundamental visualizable unit of Floating World is the ***hyperflob***, or multidimensional flying object. While many hyperflobs (or flobs for short) may appear on the screen as simple two-dimensional objects, additional dimensions may be added at any time. If you want to visualize a hyperflob, think of the old Universal Studios 3D emblem, Saturn-like, with lettering rotating around it.

All the visualizations of Floating World are based on hyperflobs and their coordinates, especially as manipulated through ***edge lists***. An edge list is a zzlist of elements placed in ascending order along a single coordinate. As an element moves in the space, its coordinates change, and its ordinal position in the edge lists may also change. Tests and interpolations on these edge lists are fundamental FW operations.

We hope to provide a user programming language, ***CLANG*** (cell language), With Clang, we hope to make it possible for the user easily to control the multidimensional behavior of user-designed hyperflobs. It will allow both casual programming (recording keystrokes to create transposable macros) and serious programming, up to and including parallel multiprocessor programs. However, it will of course not compare in efficiency to what skilled programmers do through more conventional tools.

II. FLUID MEDIA

Cells are the structure, contents are not. Contents are either conventional files (like jpg and gif) or ***fluid media***. Fluid media are digital contents which are divisible into sequentially addressable elements. Fluid media include text, audio samples, video frames, and faxes.

Fluid media are given stable addresses on ***permascrolls*** (append-and-read-only files). They may be linked to, and used in different places at the same time (transcluded), to the element level of granularity.

Fluid media are addressed and connected according to the Xanadu addressing model, managing both transclusions and content links. They are addressed referentially by lists (like the EDL in video editing). As in previous Xanadu designs, change is additive.

Links are based on the Xanadu xu88 model, where each link has two endsets and a type. Transclusions are discovered by address comparison.

III. ZZSTRUX + FLUID MEDIA + XUSTRUX

The combination of zzstructure, hyperflobs, fluid media and Xanadu connectivity allows virtually any presentation and media structure to be created and made radically interactive.

SYNCHRONIZATION

Fluid media are given permanent addresses, and cells are given permanent names and their contents locked for transmission. Thus Floating World contents may be exchanged with all contents permanently identified and no confusion among them.

The transmission medium (a mime type) will transport clusters of cells and their associated fluid media. It will be possible to send for parts which are missing. Thus complex structures may be transmitted and synchronized between machines and users without difficulty, and without forcing destruction of previous versions (like most synchronization tools).

IV. DIFFERENCES FROM THE ORDINARY COMPUTER WORLD

Here is some ways our system is different:

- "Files" are not user-significant, nor their rules of location or naming.
- No item (or project or document) needs to be named, unless the user wants. There are of course no limits as to length or characters. User names are subject to Xanadu connectivity.
- Items, projects and structures may overlap in any way the user chooses.
- The same fluid media may be used repeatedly, and you may see the different uses side by side (Xanadu transclusion).
- Previous versions and states are all reachable.
- The simple structure of every thinglet is easily user-accessible for change or study.
- All your machines can share all your data uniformly across this environment.
- Hyperflobs may be published, and may yet provide an alternative to HTML, XML and their ilk.

Who could want less?